Universidad Europea de Madrid

LAUREATE INTERNATIONAL UNIVERSITIES

UNIDAD DE CONTROL SEGMENTADA

OPTIMIZACIÓN EN LA PLANIFICACIÓN DE PIPES

Índice

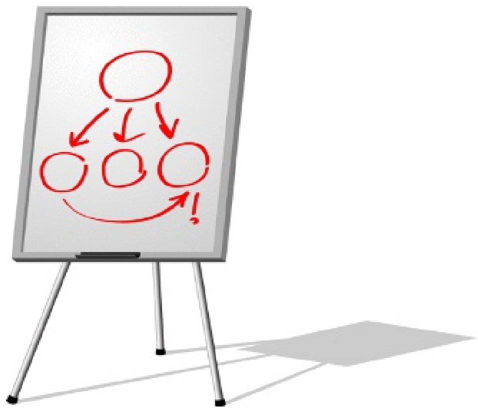
Presentación

El algoritmo para la construcción de diagramas de estado nos permite representar, de forma sencilla, todas las posibles planificaciones de un pipeline.

Pero cuando el diagrama de estados es complejo, las posibles planificaciones del pipeline pueden ser numerosas y la identificación de las planificaciones puede ser compleja:

- ¿Cómo decidir cuáles son las planificaciones potenciales de ser óptimas?
- ¿Cómo decidir cuáles hay que desechar?
- ¿Cómo saber si hay que seguir buscando más planificaciones?

En este tema aprenderemos cómo optimizar la búsqueda de ciclos de latencias que potencialmente puedan formar parte de la planificación más óptima de un pipeline.

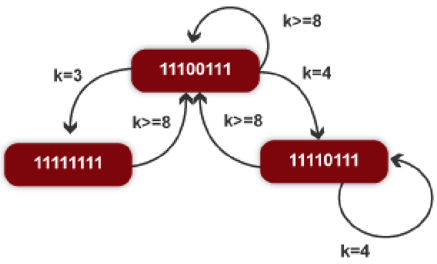


Estrategia avariciosa

El procedimiento de optimización se basará en una estrategia *Greedy* (avariciosa), que se basa en el hecho de que la primera latencia permitida tiene más probabilidad de ser óptima.

Sin embargo, la experiencia nos dice que tal circunstancia no siempre es así, pues hay ocasiones en las que merece la pena esperar varios ciclos para introducir la siguiente instrucción en el pipe y no siempre la primera latencia permitida nos lleva a la planificación más óptima.

Supongamos el siguiente diagrama de estados.



Si aplicamos la estrategia greedy, escogeremos siempre la menor latencia posible en cada paso. En este caso, el ciclo de latencias obtenido sería:

- C1 = (3,8) → LAT = 5,5

Pero si esperamos un poco, e introducimos la segunda instrucción con latencia 4, el ciclo obtenido sería más óptimo:

- **C2 = (4) → LAT = 4 → Configuración más óptima**

> En este ejemplo, se demuestra que la estrategia greedy, no siempre es la más óptima.

Lema 1

En las siguientes secciones vamos a plantear algunos Lemas relacionados con los diagramas de estados y la planificación de pipelines.

Para ello, es necesario que en este punto, definamos algunos conceptos.

Mínima latencia alcanzable (MLA)	Es la latencia media obtenida en la planificación más óptima del pipeline. Es decir, la menor latencia media que se puede alcanzar para el pipeline y la tabla de reservas que se esté estudiando.

Ciclo simple	Es aquel ciclo de latencias que solo pasa por cada estado una sola vez, excepto el primero y el último, que será el mismo estado.

Ciclo greedy	Es un ciclo simple de latencias, que se ha construido transitando siempre con la menor latencia posible en cada paso.

Lema 1

Dado un pipeline y una tabla de reservas asociada a ese pipe, siempre se cumple que **MLA ≥ Max{NM(Ei)}**.

Donde:

- NM(Ei): máximo número de marcas en una etapa de la TR
- i: 1...n etapas

En detalle Demostración

Demostración

Sabemos:

- Que la ocupación media de una etapa se calcula como O(Ei) = NM(Ei) / LAT.
- Que O(Ei) ≤ 1, por lo que NM(Ei) / LAT ≤ 1.

El máximo valor que puede tomar la relación NM(Ei) / LAT se obtendrá cuando:

- NM(Ei) sea máximo = Máximo número de marcas en una etapa de la TR.
- LAT sea mínimo = MLA.

El peor caso de la igualdad será este caso:

Max{NM(Ei)} / MLA ≤ 1 → **MLA ≥ Max{NM(Ei)}**

Lema 2

Para todo diagrama de estados, se cumple:

- Si $\exists$ un ciclo C cuya LAT = L → $\exists$ un ciclo simple S cuya LAT ≤ L.

De este lema podemos extraer la siguiente conclusión: los ciclos óptimos son siempre ciclos simples. Esta conclusión reduce la búsqueda de configuraciones óptimas a los ciclos simples, pero aun así el número de ciclos posibles puede ser todavía muy elevado.

Demostración

Es lógico que el ciclo óptimo sea simple, porque si fuera complejo implicaría que en un mismo ciclo se utilizasen dos estrategias distintas. El ciclo de latencias se construye como la combinación de dos ciclos simples distintos.

Cuando utilizamos un ciclo de latencias construido como combinación de dos ciclos simples, pueden ocurrir dos cosas.

Los dos ciclos simples tienen la misma latencia media	Por tanto, el ciclo no simple resultante también tiene la misma LAT. En este caso LAT = L1 = L2. Siendo LAT la latencia media del ciclo no simple, y siendo L1 y L2 las latencias medias de los ciclos simples incluidos en el ciclo no simple.
Uno de los dos ciclos simples tiene menor latencia media	En tal caso, la LAT del ciclo no simple será la media de las latencias medias L1 y L2, por lo que el resultado LAT será peor que en el ciclo más óptimo entre los dos primeros.

En forma genérica, podemos decir que la latencia media de un ciclo no simple, cumple:

LAT ≤ Max {Li}, i = 1...m

Siendo m el número de ciclos simples dentro del ciclo no simple.

Lemas 3 y 4

Lema 3

Dado un ciclo greedy C que tiene una latencia media LAT, se cumple:

- Si LAT = Max{NM(Ei)} → C es un ciclo óptimo.

Demostración del lema 3

Como O(Ei) = Max{NM(Ei)} / LAT:

- Si LAT = Max{NM(Ei)} → Max{NM(Ei)} / LAT = 1 → O(Ei) = 100%.
- Si una etapa tiene ocupación media de 100%, significa que el ciclo C es óptimo.

Lema 4

Para todo ciclo simple greedy, C con latencia media LAT, se cumple:

- LAT ≤ Número de unos en el vector de colisión inicial.

¿Te atreves a demostrarlo?

Búsqueda de ciclos greedy

Para simplificar la búsqueda de la planificación óptima, y apoyándonos en los lemas anteriores, vamos a reducir el amplio número de posibles secuencias de latencias, a **ciclos de latencias que contengan un ciclo greedy**.

Nuestro procedimiento de búsqueda de ciclos en un diagrama de estados, se reducirá a la búsqueda de ciclos greedy.

Para facilitar esta tarea, enunciaremos el **algoritmo de búsqueda de ciclos greedy.**

1. Elegir un estado E no analizado.

2. Seguir los arcos de latencia mínima hasta que:

a) Se regresa al estado E.

b) No existen más arcos para transitar.

3. a) En el caso 2a, el ciclo hallado es greedy. Añadir a la lista de ciclos greedy y eliminar sus estados del diagrama.

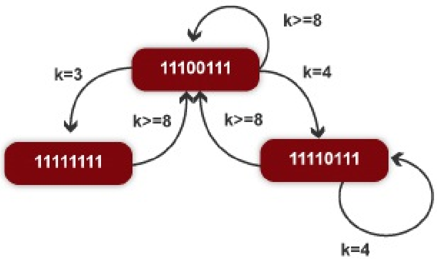
b) En el caso 2b, eliminar el estado E.

4. a) Si quedan estados sin analizar ni eliminar, ir a paso 1.

b) En caso contrario, hemos terminado. El resultado de la búsqueda se encuentra en la lista de ciclos greedy.

Ejemplo

Para seguir con el ejemplo anterior, y dado el diagrama de estados que se muestra a la derecha de estas líneas, calculemos sus ciclos greedy aplicando el algoritmo de búsqueda de ciclos greedy.

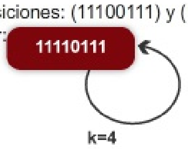


Primera iteración de búsqueda de ciclos greedy

1. Se puede empezar por cualquier estado. Escogemos el estado inicial **E = 11100111**.
2. Si transitamos con las latencias mínimas, obtenemos el siguiente ciclo: C1 = (3,8) LAT = 5,5
3. Añadimos C1 a la lista de ciclos greedy:

Ciclos greedy = {C1 = (3,8)}

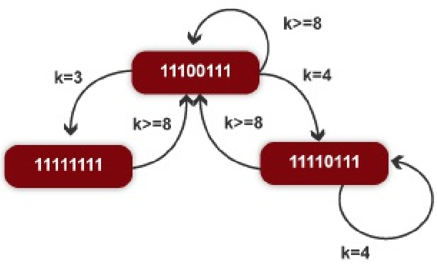
Y eliminamos los estados participantes en C1 y sus transiciones: (11100111) y (11111111).
Nos queda el siguiente diagrama de estados por analizar:



4. Como queda un estado por analizar. Volvemos al paso 1.

1/2

Para seguir con el ejemplo anterior, y dado el diagrama de estados que se muestra a la derecha de estas líneas, calculemos sus ciclos greedy aplicando el algoritmo de búsqueda de ciclos greedy.



Segunda iteración de búsqueda de ciclos greedy

1. Escogemos el único estado que nos queda por analizar: **E = 11110111**.
2. Si transitamos con las latencias mínimas, obtenemos el siguiente ciclo: C2 = (4) LAT = 4.
a) Añadimos C2 a la lista de ciclos greedy:

Ciclos greedy = {C1 = (3,8), C2 = (4)}

Y eliminamos el estado participante en C2: (11110111). Ya no quedan estados por analizar.

b) Como no quedan más estados por analizar, estos son todos los ciclos greedy hallados en el diagrama de estados:

Ciclos greedy = {C1 = (3,8) , C2 = (4)}

2/2

Optimización en la planificación

Para optimizar el tiempo empleado en planificar una unidad de control segmentada no necesitamos estudiar todas las posibles planificaciones que se derivan de un diagrama de estados, sino que nuestra búsqueda de la planificación óptima se enfocará solamente a la búsqueda del ciclo greedy más óptimo en el diagrama de estados.

Así pues, y dada una tabla de reservas T asociada a una unidad de control segmentada, definimos nuestro **nuevo algoritmo optimizado de planificación de pipelines**:

1. Aplicar el algoritmo de generación del diagrama de estados para la tabla T:

- Entrada: tabla de reservas T.
- Resultado: diagrama de estados D.

2. Aplicar el algoritmo de búsqueda de ciclos greedy:

- Entrada: diagrama de Estados D.
- Resultado: lista de ciclos greedy L.

3. Para cada ciclo greedy Ci L:

- Calcular latencia media Ci → LATi.
- Escoger el ciclo greedy Cm con menor latencia media LATm.

LATm = min {LATi} = MLA (Mínima latencia alcanzable).

- Entrada: lista de ciclos greedy L.
- Salida: ciclo greedy Cm con LATm = MLA.

4. Construir una secuencia de latencias desde el estado inicial (VCini) hasta Cm = (Lm1, Lm2, ... , LmN):

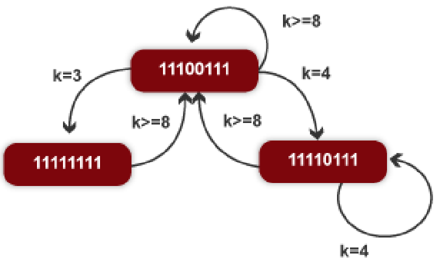
Secuencia óptima: Sm = <L1, L2, ..., Lm1, Lm2, ... , LmN... >

- Entrada: ciclo greedy óptimo (Cm), y diagrama de estados (D).
- Salida: planificación óptima del pipeline Sm.

Ejemplo de optimización

Siguiendo con nuestro ejemplo, si aplicamos el algoritmo optimizado de planificación de pipelines, ya tenemos desarrollados casi todos los pasos.

Diagrama de estados



Lista de ciclos greedy

Ciclos greedy = {C1 = (3,8) , C2 = (4)}

Latencias medias

- LAT (C1) = 5,5
- LAT (C2) = 4 → C2 es el ciclo greedy más óptimo.

Construcción de la planificación óptima.

El ciclo greedy C2 = (4) se obtuvo desde el estado 11110111.

Por tanto, la secuencia óptima se calcula empezando desde el estado inicial hasta 11110111:

Sm = <4, 4,4,4,4,4,...>

Podemos representar la planificación óptima como un ciclo de latencias: **Cm = (4)**.

Resumen

El algoritmo optimizado de planificación de pipelines nos permite automatizar el procedimiento completo de planificación de pipelines.

Además, permite reducir enormemente el tiempo empleado en la búsqueda de secuencias de latencias a estudiar para poder encontrar la planificación más óptima de un pipeline.

Este algoritmo se basa en la premisa de que la planificación más óptima del pipe contiene algún ciclo greedy.

Por tanto, para encontrar la planificación óptima basta con enfocarnos a la búsqueda de ciclos greedy dentro del diagrama de estados y obtener la secuencia con menor latencia media.